

Software Engineering

Introduction

What is software?

- A program is a number of programming language code that performs certain task(s). Only one programmer can design, implement and test the program.
- A software is large complicated program with hundreds of lines of code. It needs the cooperation of many individuals to create a SW: Analysts, designers, programmers, testers, maintenance engineers, and others. A SW engineer is required for organization.
- SW: Computer program and associated documentation

What is software?

- Software products may be developed for a particular customer or may be developed for a general market
- Software products may be:
 - **Generic** - developed to be sold to a range of different customers
 - **Bespoke (custom)** - developed for a single customer according to their specification

Difference between software engineering and system engineering

- System engineering is concerned with all aspects of computer-based systems development including: hardware, software and process engineering.
- Software engineering is part of this process.
- System engineers are involved in system specification, architectural design, integration and deployment

SW crisis

- After the appearance of the new generation of computers (50's) with higher speed.
- Program Scaling: adding new lines to the same program to add new functions.
- Errors found, and program crash.
- Possible sources of error: reusing the names of the variables and functions, hard to tract the sequence of the data flow inside program,...

Software Engineering

- **Software engineering is an engineering discipline which is concerned with all aspects of software production.**
- **It is the science concerned with building large and complicated SW systems in a cost effective way.**
- **It is the application of systematic, disciplined approach for development, operation and maintenance of SW.**

Software Engineer

- Software Engineer (SWE) is the person responsible for organizing the work of the group of persons whom will build the SW.
- Software engineers must have:
 - Technical aspects: (programming, mathematics, maintenance, test, analysis)
 - Non technical aspects: (communication ability both oral and written, management, personality,)
- SW cost Vs HW cost

Challenges facing software engineering

1- Heterogeneity:

- Some of the software systems are allocated on networks as distributed system which will be used by different types of users working on different types of computers and operating systems.
- This is called heterogeneity of users. This kind of problem must be solved by developing new techniques for building software that can deal with this heterogeneity without causing problems.

Challenges facing software engineering

2- Delivery:

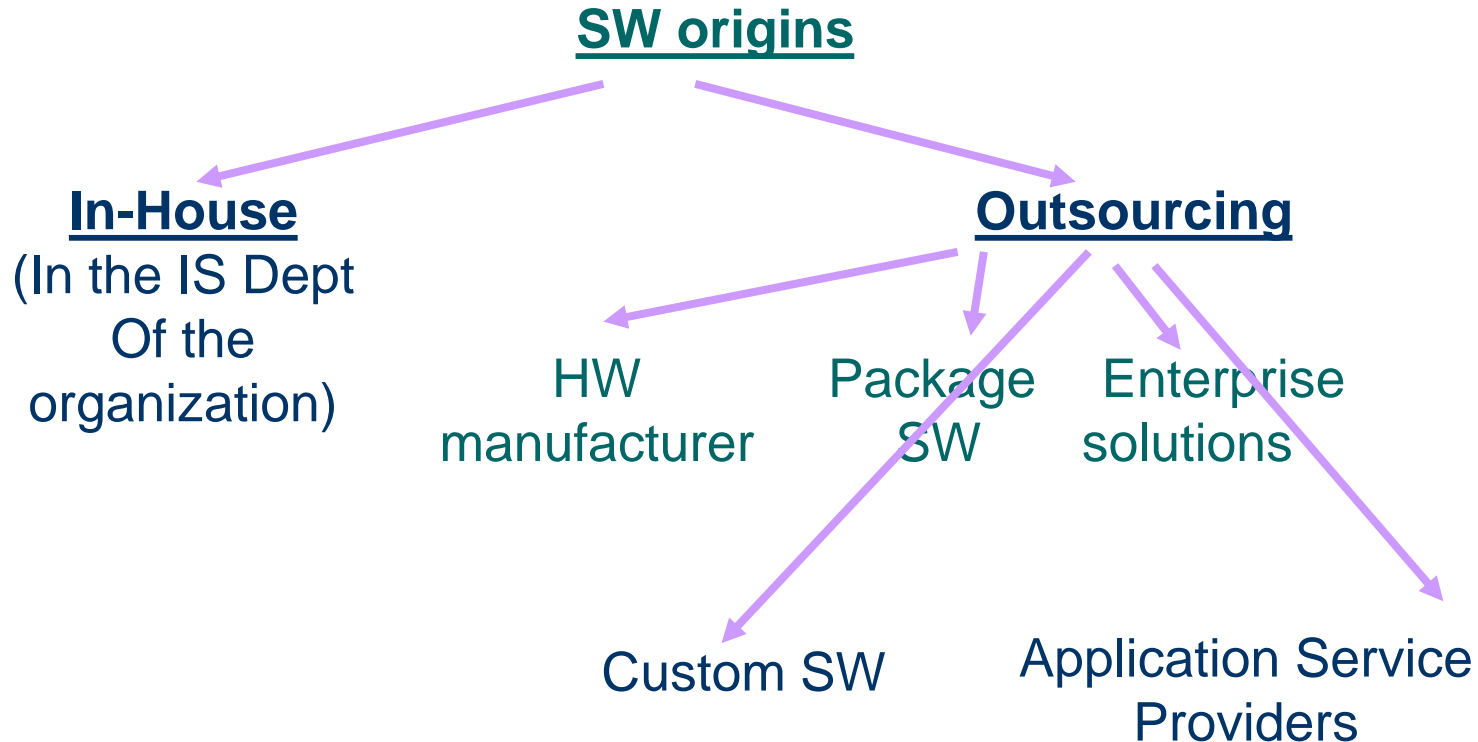
- As you will see, the development of software system contains many steps that consume time.
- Due to the rapid change of business and users needs from the software, the steps of developing the software must be shorten to cope with this change.
- This should not be done on the account of the software quality.

Challenges facing software engineering

3- Trust:

- Software application are penetrating in all our daily live starting from the bank software, the mall software, the university software and even the e-mail software.
- If the user does not trust these softwares to do their work efficiently, then they will be of no use to him.
- Thus, we need to develop techniques to ensure the correctness, integrity, completeness, confidentiality etc. of these softwares in order to fulfill the user's trust.

The Origins of Software



In-House Development

- The company has its own Information system dept. which is capable of producing its own applications.
- The whole Software Development steps are performed in special dept. at the company.
- No need for external help.

Outsourcing

- The practice of turning over responsibility of some or all of an organization's information systems applications and operations to an outside firm.
- One organization develops or runs (or both) a computer application for another application.
- One extreme option: a firm that develops and runs your application on their own computers, all you have to do is to provide input and take output (Ex. Employee Payroll for small company)

- Another option: you hire a company to run your applications at your site on your computers.
- 80% of companies use a kind of outsourcing.

● Why Outsourcing?

- Cost effective solution
- Possess knowledge & skills not found internally
- Freeing up internal resources
- Reduce time to market
- Increase the revenue
- Increase process efficiencies

Hardware Manufacturers

- IBM HW company has the highest sales in SW (66,451 Millions) in 2007 report
- Selling not traditional application SW but SW for web servers and application servers, middle (SW that links one set of SW services to another).
- Compaq

Packaged SW Producers

- Microsoft (Windows, office)
 - Oracle (DBMS)
-
- These products are generic (for general users) and called off-the-shelf SW.

Advantages	Disadvantages
Increased reliability and functionality	Must accept existing features
Often lower initial cost	Limited customization
Already in use by other firms	Uncertain future of vendor
Help and training comes with software	Less ownership and commitment

Custom SW Companies

- Company needs IS but does not have expertise or personnel to develop In-House and there is no suitable off-the-shelf product.
- It will consult Custom SW Company (ex. EDS, Acceture, CSC, Compaq) to develop a special (custom) SW for this company.

Advantages	Disadvantages
Specific response to specialized business needs	May be significantly higher initial costs compared to others
Innovation may give competitive edge	Ongoing maintenance
In-house staff for maintenance	Necessity of hiring or working with a development team
Pride of ownership	

Application Service Providers

- A company rents or licenses an application from a third-party provider who runs the application at remote site.
- Users have access to the application through Internet or VPN (Virtual Private Network).
- The application provider buys, installs, maintain the application and makes it available for others for a certain price, thus they are called application service providers (ASP).

Enterprise Solutions Software

- Companies around the world are requesting softwares for all their aspects of work like accounting, distribution, manufacturing, human resources,....
- That was the reason that enterprise solutions (called enterprise resource planning ERP systems) are presented.
- The difference between this approach and the traditional approach of programming is that we will need a separate system for each function while here all the functions are completing each other.

Open-source software

- Open-source software is a free software that you can obtain without any charge from your side.
- Through this kind of software, you have the running version of the software free or even you can get the source code itself.
- Examples of such software are: Linux (operating system), MySQL (database management system), and Firefox (internet browser).
- You can also get codes for encryption, compression, scientific problems,.... from these sites that hosts such free software. You can change in the code as you like without any problems.

Criteria for Choosing Off-the-Shelf Software

- **Cost** - In-House versus purchased
- **Functionality**
 - Mandatory, essential and desired features
- **Vendor Support**
 - Installation -Training
 - Technical Support
- **Viability of Vendor**
- **Flexibility**
 - Ease of customization
- **Documentation**
 - User documentation
 - Technical documentation
- **Response Time**
- **Ease of Installation**

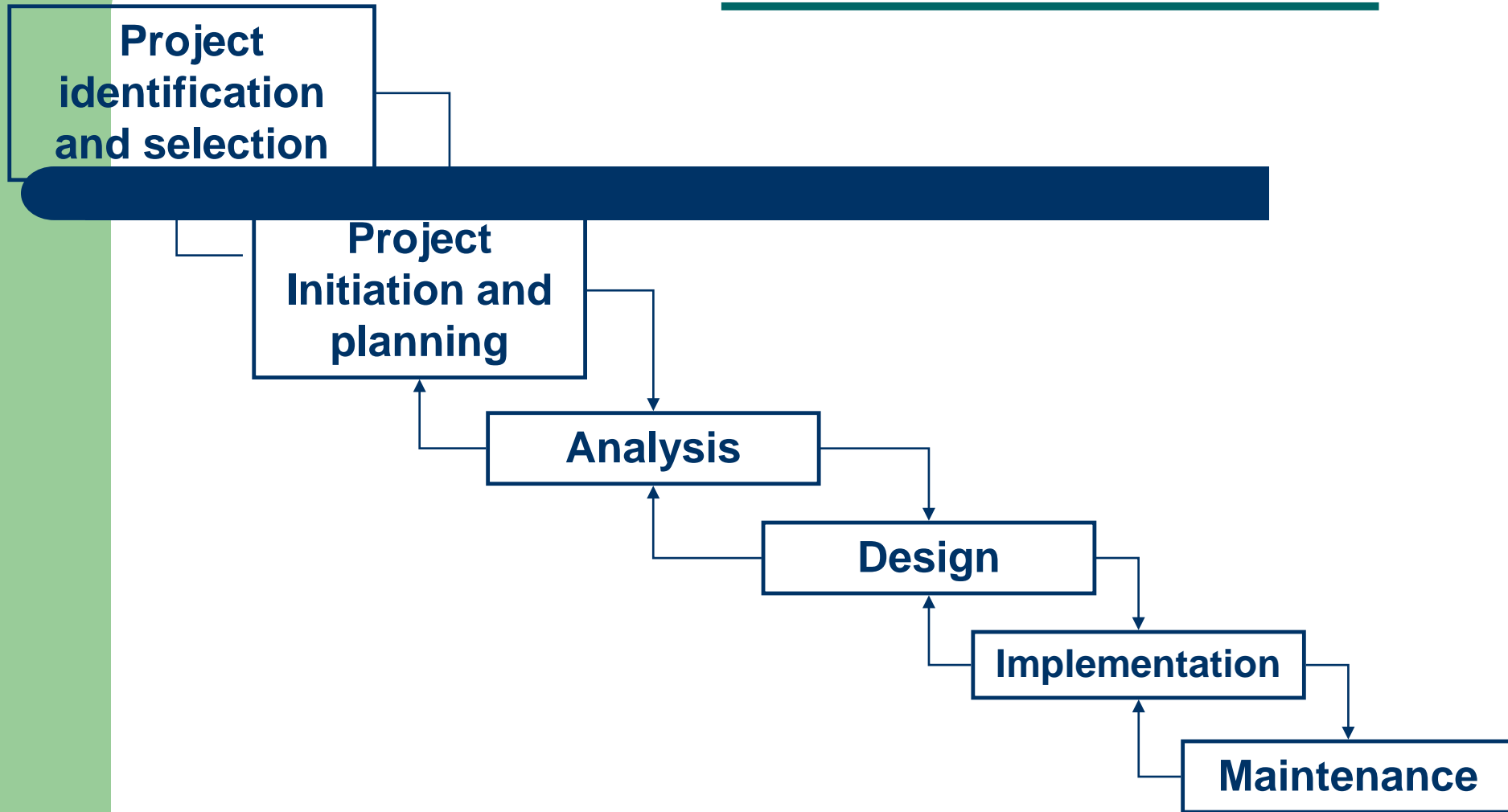
Validating Purchased Software Information

- **Information from vendor**
 - **Request for proposal**
 - A document provided to vendors to ask them to propose hardware and system software that will meet the requirements of your new system
- **Software evaluation period**
- **Customer references from vendor**
- **Independent software testing service**

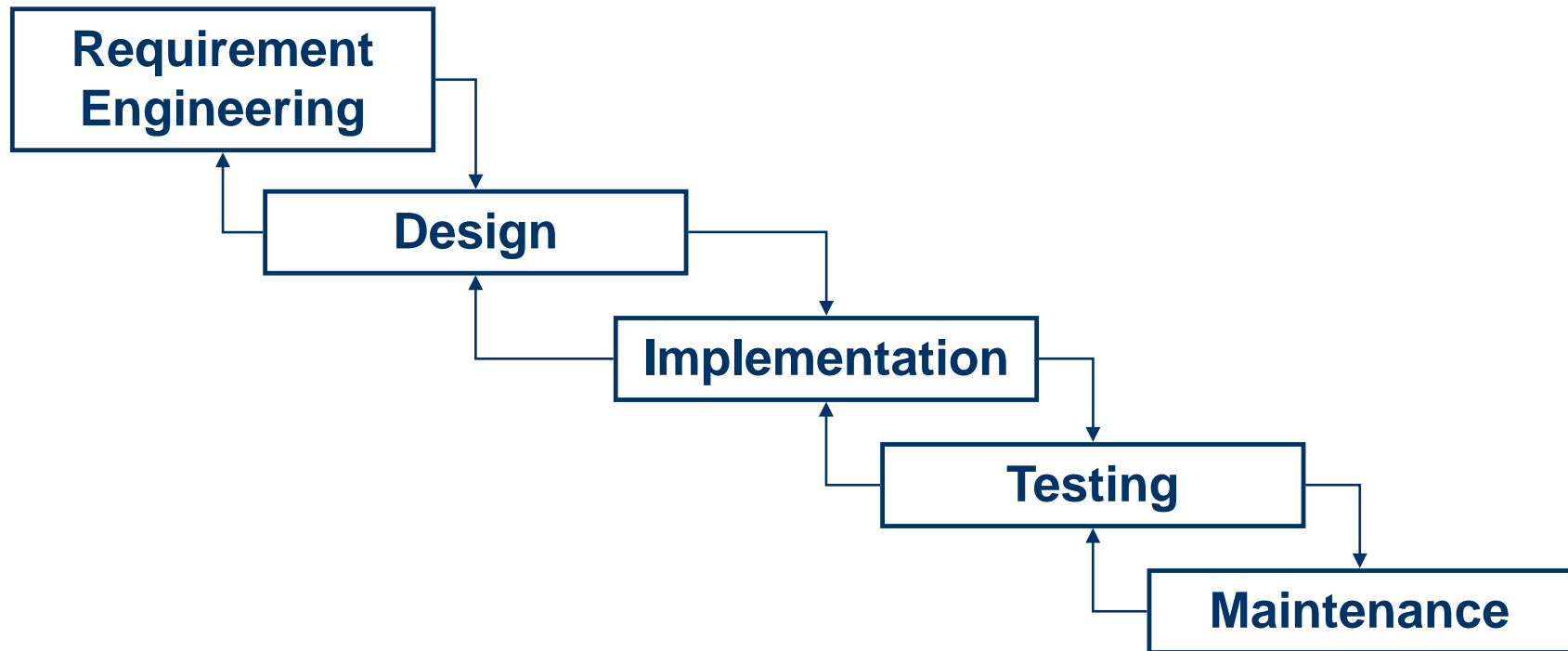
SDLC (System Development Life cycle)

- They are the steps required to develop SW. We have many choices according to the case:
 - Waterfall model
 - Prototyping
 - Incremental development
 - Integration from reusable components
 - Spiral model

1- Waterfall Model



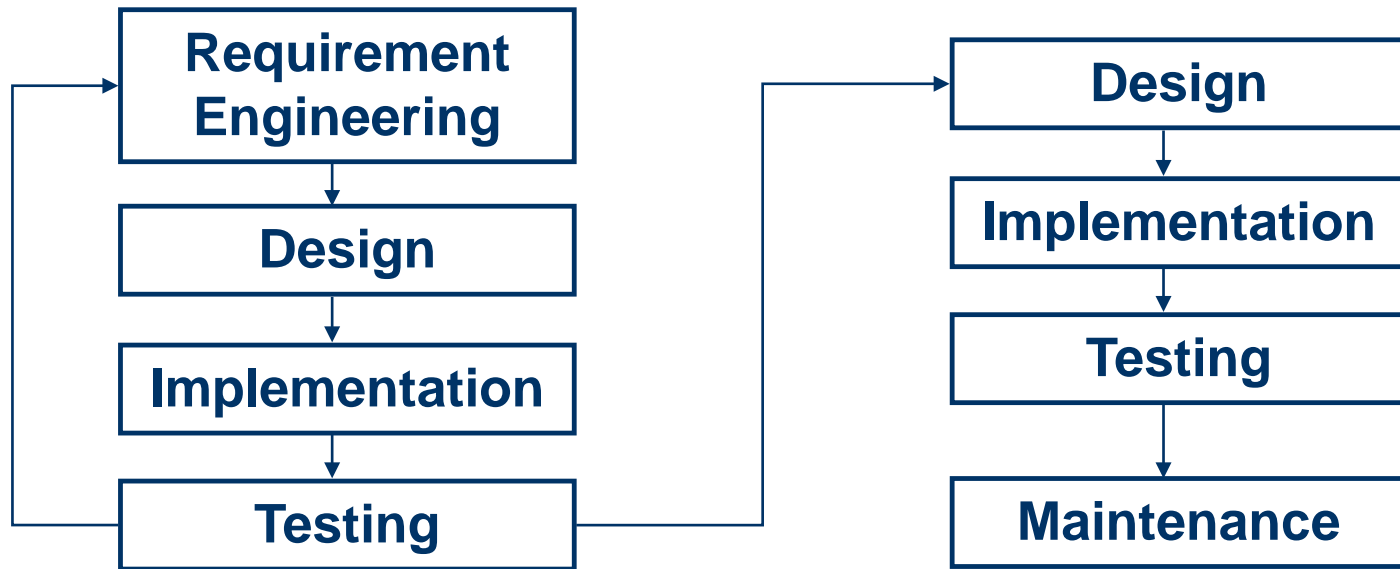
Waterfall Model (compact)



2- Prototyping

- In systems where the user requirements are unclear or newly developed systems, we better make a prototype of the system before making the final version.
-
- In other products, like cars or telephones, we can make a number models (prototypes) with different organizations. We choose one of them after testing them to go to the production line.
 - This is not applicable in SW, as the larger cost in the other products are in making copies of the chosen model.
 - But, in SW making copies is free !!!!!
 - To make a SW prototype, we can:
 - 1- Make one complete working part of the SW which will be repeated for the other functions of the SW.
 - 2- Develop a SW with less functions (or non working functions).

Prototyping



Prototype

**Final Version
of SW**

Prototyping

Types:

- **Throwaway Prototype:** objective is to understand the system requirements. Should start with poorly understood requirements
- **Evolutionary Prototype:** evolve the prototype by adding new requirements or change of them generating new version of prototype. (much more used).

Problems:

- Lack of process visibility
- Systems are often poorly structured
- Need of special skills (e.g. how to make such prototyping)

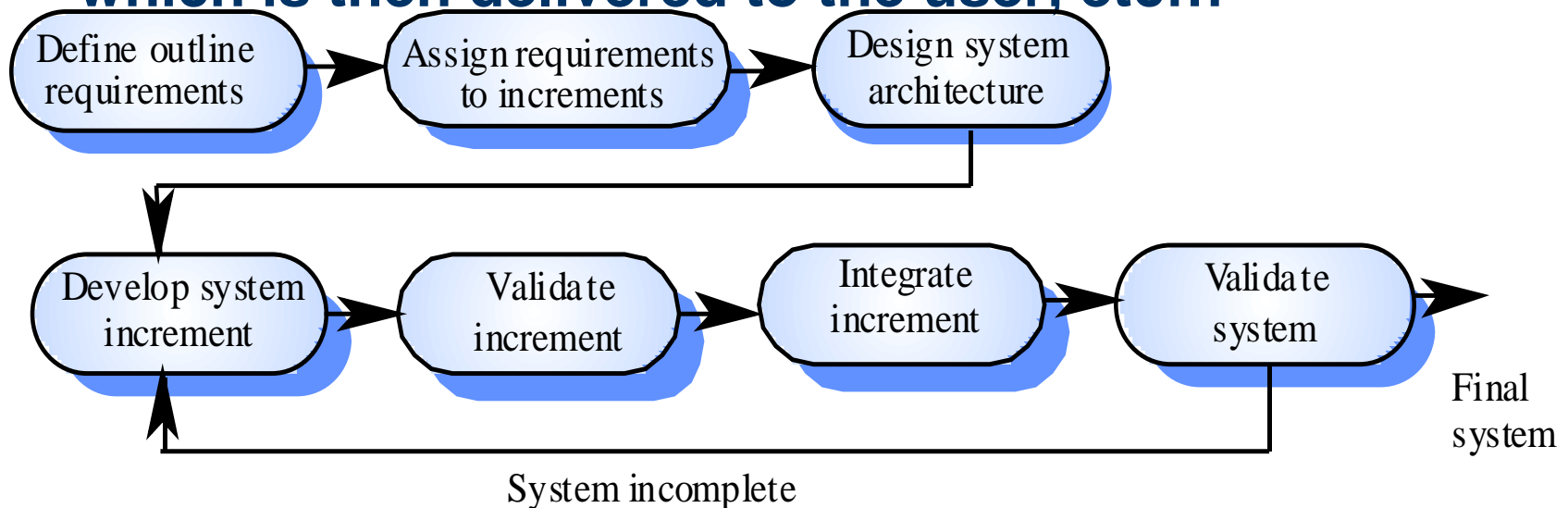
Prototyping

- **This model is better used when:**
 - **The user requirements are not clear**
 - **For small or medium-size interactive systems (The user interface is important)**
 - **For parts of large systems (e.g. the user interface)**
 - **For short-lifetime systems**

3- Incremental Development

- In prototyping, the final version of the SW is the only one delivered.

- In Incremental development, a small, yet working version of the SW is delivered to the users.
- Then step by step we add or modify functions into the available version generating a new version which is then delivered to the user, etc...



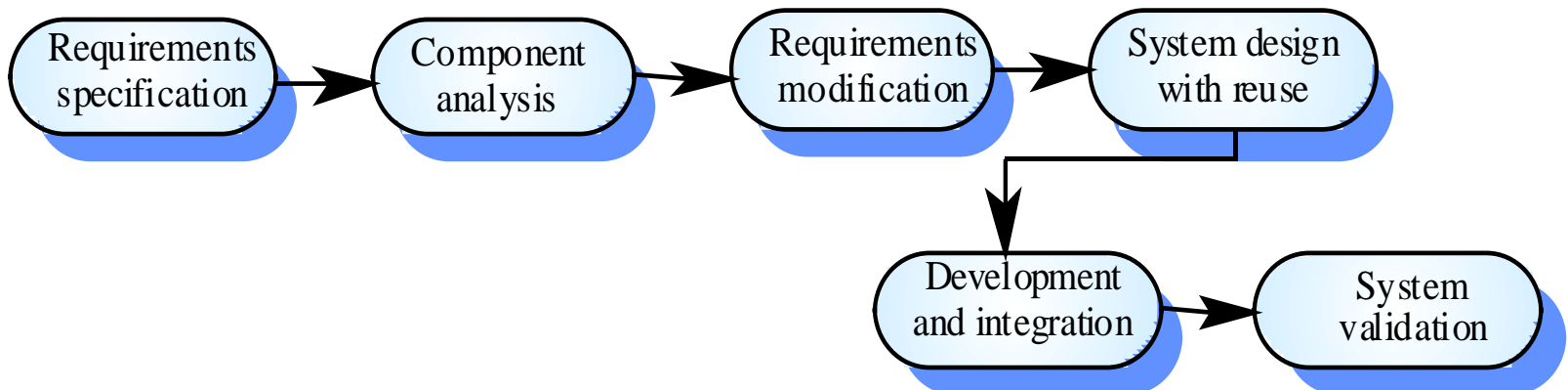
Incremental Development

- Rather than deliver the system as a single delivery, the development and delivery is broken down into increments with each increment delivering part of the required functionality.
- User requirements are prioritised and the highest priority requirements are included in early increments.
- Once the development of an increment is started, the requirements are stopped.
- The requirements for later increments can continue to evolve.

4- Integration from reusable components

- This model is highly in use now specially in big companies.

- Existing components of previous developed SW are integrated to generate new SW.
- This minimizes the cost and time required to build SW from zero point.

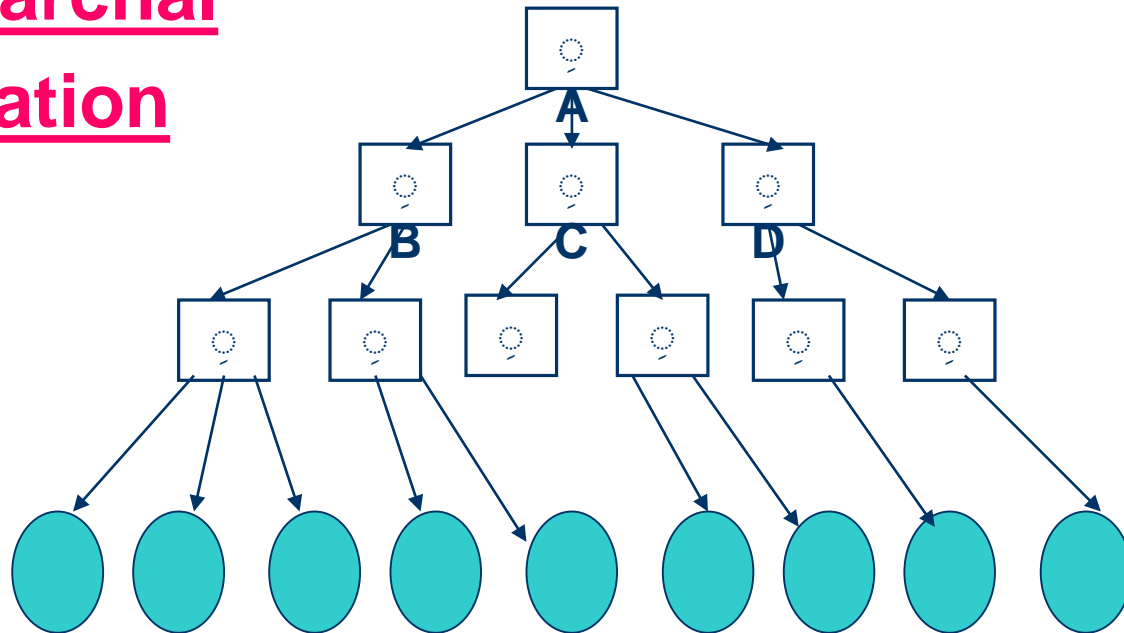


5- Spiral Model

- The spiral model integrates both the prototype model and the incremental development model to generate the SW. It can also integrate the maintenance activities.
- In each round of the spiral, the following activities are performed
 - 1- Identify a problem (maintenance), new step (incremental) or new requirement (prototype).
 - 2- Solve the problem by following the rest of the steps of the SDLC.

Team Organization

- 1-Hierarchical Organization



Suitable for big companies

1-Hierarchal Organization

Advantages:

- Well organized

Disadvantages:

- Takes a long time for orders to go from up to down or complaints from down to up (may get lost or stalled)
- The lower levels are the actual workers, and are not allowed to make decisions.

2- Matrix organization

- Suitable for medium and small size companies.
- Each employee has a matrix which describes his role in the different available projects.

Table for worker 1

Project ID	Analysis	Design	Test	Graphics	Database
Project 1	x		x		x
Project 2	x	x		x	

2- Matrix organization

Advantages:

- 1- Use all the person capacity
- 2- Save human resource, and thus save money.

Disadvantages:

- 1- No specialization for each worker (no professionalism). Thus, the quality of his work is not high.
- 2- Any worker may have more than one project running at the same time, i.e. working all the time, so he can make mistakes

2- Matrix organization

To solve this problem:

- We can specify two or more specialties for each worker, so his work will be better.

3- Chief in programmers

- Used in small companies
- It consists of a group of equal skills programmers, under the control of a chief programmer (has higher skills and experience).
- A secretary is used for doing the paper work.
- All decisions comes from the chief programmer.

4- SWAT Team

- **SWAT = Skilled Wirth Advanced Technology**
- **It is a group of 5-6 programmers having the same and efficient skills.**
- **The decisions are taken through votes.**